Fair Learning-to-Rank from Implicit Feedback

Himank Yadav*
himankyadav1@gmail.com
Cornell University
Ithaca, NY, USA

Zhengxiao Du*
duzx16@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Thorsten Joachims
tj@cornell.edu
Cornell University
Ithaca, NY, USA

ABSTRACT

Addressing unfairness in rankings has become an increasingly important problem due to the growing influence of rankings in critical decision making, yet existing learning-to-rank algorithms suffer from multiple drawbacks when learning fair ranking policies from implicit feedback. Some algorithms suffer from extrinsic reasons of unfairness due to inherent selection biases in implicit feedback leading to rich-get-richer dynamics. While those that address the biased nature of implicit feedback suffer from intrinsic reasons of unfairness due to the lack of explicit control over the allocation of exposure based on merit (i.e, relevance). In both cases, the learned ranking policy can be unfair and lead to suboptimal results. To this end, we propose a novel learning-to-rank framework, FULTR, that is the first to address both intrinsic and extrinsic reasons of unfairness when learning ranking policies from logged implicit feedback. Considering the needs of various applications, we define a class of amortized fairness of exposure constraints with respect to items based on their merit, and propose corresponding counterfactual estimators of disparity (aka unfairness) and utility that are also robust to click noise. Furthermore, we provide an efficient algorithm that optimizes both utility and fairness via a policy-gradient approach. To show that our proposed algorithm learns accurate and fair ranking policies from biased and noisy feedback, we provide empirical results beyond the theoretical justification of the framework.

1 INTRODUCTION

Implicit feedback from user behavior (e.g., clicks, dwell times, purchases, scroll patterns) [29, 38] is an attractive source of training data for learning-to-rank (LTR). It is not only abundant in most application settings, timely, and easier to collect than expert relevance judgments; it also gives all users a voice in what the system learns. However, does this participatory nature of implicit feedback automatically ensure that the learned ranking policies are fair? We argue that both intrinsic and extrinsic factors can lead to unfair ranking policies if left unchecked.

Intrinsic factors are internal to the ranking system and stem from the allocation policies that underlie the design of the system. Specifically, when deciding which ranking is presented to a user, the ranking system makes an explicit choice of how much exposure each ranked item receives – where higher-ranked items receive more exposure and thus more opportunity (e.g., to be purchased or read) [17, 19].

Merit-based exposure allocation [8, 9, 42, 43] argues that the fraction of exposure each item receives should be linked to its merit (i.e., relevance) and that there should be a well-defined relationship between merit and exposure allocation. Following this reasoning, a policy is considered unfair if it does not allocate exposure based on this defined relationship with merit, and it has been shown that conventional ranking policies that are merely trained to optimize the utility to the users can be unfair in their allocation of exposure to the items [8, 9, 42, 43].

Extrinsic factors that can lead to unfairness typically manifest themselves as biases in the training data. In particular, in order to implement the merit-based allocation of exposure, it is important to have unbiased estimates of merit.

Unfortunately, implicit feedback data is typically biased [28]. One such bias, called position bias, exists because feedback collected by these systems is biased towards items ranked highly in the past [30]. Under one-sided feedback like clicks, highly-ranked items receive more clicks (i.e., positive feedback) due to the increased attention they receive, which further skews the ranking system and affects future rankings. This results in a dynamic amplification of position bias and leads to the well-studied phenomena of rich-get-richer [1, 28, 30, 41] and few-get-richer [21].

In this paper, we present a framework – called FULTR (Fair Unbiased Learning-to-Rank) – for designing fair LTR algorithms that address both intrinsic and extrinsic sources of unfairness. Specifically, we propose the first method and training algorithm that can enforce merit-based exposure constraints while at the same time debiasing logged implicit feedback data. To address intrinsic fairness, we show that existing fairness constraints [9, 42, 43] cannot be applied under biased feedback, and we define a novel type of amortized fairness-of-exposure constraint for group-based fairness. For this type of fairness constraint, we derive counterfactual estimators [30, 44] that can provably correct the position bias that leads to rich-get-richer dynamics. The latter addresses extrinsic fairness due to the selection bias that is induced by the presented rankings. To make the proposed framework operational and practical, we show how to search the space of fairness-constrained ranking policies via a policy-gradient algorithm. The evaluation of FULTR rests both in its theoretical justification as well as in an extensive empirical evaluation on real-world datasets. We find that FULTR can effectively optimize utility and fairness over a range of settings even when trained with biased and noisy feedback.

2 RELATED WORK

There have been numerous approaches to defining fairness in different areas of machine learning, including online learning [23], classification [2, 33], regression [7, 39], and multi-armed bandits [35]. We focus on fairness in the relatively under-explored domain of LTR, which has only recently caught attention despite its substantial implications in a broad range of real-world applications. To structure the discussion, we follow the distinction of extrinsic and intrinsic sources of unfairness introduced above.

*Equal contribution

Concerning intrinsic fairness, several methods have followed the concept of demographic parity. Demographic parity does not consider merit, but merely enforces a proportional allocation between groups. This is achieved by either reducing the difference in occurrences of different groups on a subset of the rankings [46] or by placing a limit on the number of items from each group in the top-k positions [14, 22, 47].

Merit-based fairness of exposure proposed by Singh and Joachims [42] makes exposure allocation not merely dependent on group size, but it allocates exposure to items based on their merits. The algorithm in [42] shows how this can be formulated as a linear program over doubly-stochastic matrices. From its solution, a stochastic ranking policy can be derived via the Birkhoff-von Neumann decomposition. An alternative is the integer-programming algorithm of Biega et al. [9], which dynamically optimizes amortized individual fairness of exposure. However, both of these methods do not involve learning and assume full knowledge of the true relevance labels for all items to be ranked. In practice, true relevance labels are not available, and one needs to use other learning methods to impute relevance labels at prediction time. This leads to a two-step process, where the first regression step is unaware of fairness considerations and fairness is only introduced during post-processing. We find that this can lead to situations where fairness post-processing can not recover from lousy regression estimates. Our FULTR framework, on the other hand, performs end-to-end learning with fairness, and it is not limited in this way.

More recently, Zehlike and Castillo [48] proposed an LTR method that incorporates an exposure-based ranking loss along with a fairness regularizer. However, this algorithm is limited to a fairness metric that only considers the top-1 position in each ranking but not how items below are ranked. This limitation was removed in the work by Singh and Joachims [43], which proposes an end-to-end LTR method that optimizes both utility and fairness constraints simultaneously for the full ranking. However, this method does not consider extrinsic sources of unfairness, and it assumes that expert-labeled data for all items is available during training. This is a weaker assumption than the one made in [9, 42], but it still does not apply to real-world settings where implicit feedback is used for training. We overcome this limitation by proposing novel counterfactual estimators that debias extrinsic unfairness in implicit feedback data due to position bias, presenting the first end-to-end LTR algorithm with fairness-of-exposure guarantees where expert relevance labels are not required during training or evaluation.

Another approach to defining intrinsic fairness in rankings without considering exposure as the key criterion focuses on pairwise comparisons [8, 34]. This approach is more suitable for applications where the ranking is not presented as part of an interactive system, such that exposure is not well-defined. Fairness criteria based on the pairwise comparison between items count all swaps in the ranking equally and do not reflect that swapping items at the top has a stronger effect on exposure than doing so at lower positions.

Switching to extrinsic sources of unfairness, most work on traditional LTR algorithms [11–13, 27] has assumed that unbiased relevance judgments by experts are available. However, the field of information retrieval has long been conscious of the biases inherent to implicit feedback and its effect on the ability to rank well [17, 19, 24, 27, 28]. In particular, various studies have demonstrated the presence of position bias, where the quantity and quality of the feedback depend on the rank at which an item is presented.

One approach to modeling and removing position bias is generative click modeling as surveyed in [16]. Click models provide a way of modeling bias and estimating relevances of the items being ranked. These relevance estimates can then be used as a substitute for expert labels in LTR. Click models [10, 15, 16] typically treat relevance as a latent variable, and perform inference by the maximizing log-likelihood of clicks. Unfortunately, most click models suffer from the limitation of requiring large amounts of repeat impressions for individual query-item pairs, which makes them inapplicable to tail queries.

Recent and more direct approaches to dealing with position bias are counterfactual learning methods as proposed in [30, 44]. These methods use techniques from causal inference like inverse propensity score (IPS) weighting [40] and do not require repeated queries or latent-variable inference. Instead, they directly optimize over a debiased utility objective while incorporating click data in a principled fashion. Additional algorithms [5, 25] that jointly estimate the propensities and optimize the performance have been proposed. Unlike our proposed FULTR framework, existing counterfactual learning methods do not control for intrinsic unfairness.

Our work focuses on position bias, but another extrinsic source of bias is trust bias [28]. It captures that position affects not only attention but also the users' valuation of the items due to the trust they place into the ranking system to bring relevant items to the top. In this paper, we focus on addressing position bias, but conjecture that trust bias can be incorporated as well using the debiasing techniques proposed by Agarwal et al. [3].

The only existing method that aims to address both intrinsic and extrinsic sources of unfairness in rankings is [6]. However, it requires interactive experimental control in the form of an online algorithm, whereas FULTR can reuse logged implicit feedback data from past interactions for learning. This makes them incomparable since they rely on fundamentally different access to data. To the best of our knowledge, no other existing method addresses fairness in rankings while directly learning from logged implicit feedback.

3 LEARNING FAIR RANKING POLICIES FROM IMPLICIT FEEDBACK

We now present our counterfactual framework for learning fair ranking policies from biased implicit feedback. Our framework builds upon the merit-based fairness of exposure approach [42] and provides the first learning algorithm for enforcing merit-based fairness with logged implicit feedback. We start by defining the problem of learning fair ranking policies from implicit feedback in the context of empirical risk minimization (ERM). This identifies the need for an unbiased estimator of utility despite the biases in implicit feedback, so we propose a counterfactual estimator for this problem. Furthermore, we propose an amortized fairness constraint and the corresponding disparity measure, for which we also provide an unbiased counterfactual estimator. We then analyze the connection between our disparity measure and the Disparate Treatment constraint as proposed in [42] and amortized notions of fairness from [9]. Finally, we analyze the effect of click noise on

the disparity measure and propose a noise-corrected estimator of disparity.

3.1 Learning Ranking Policies via ERM

Learning to rank (LTR) is the problem of learning a ranking policy π from a training set $\mathbf{Q}$ of queries. We assume that queries are drawn i.i.d., $q \sim P(q)$. Each query q has a set of candidate items d^q that need to be ranked. Items can represent a wide variety of things depending on the application, e.g., web-pages in the context of a search engine, jobs in the context of a job board, and movies in the context of a streaming service. Each item is associated with a feature vector $x_q(d)$ that describes the match between item d and query q.

In the so-called full-information setting, it is assumed that all relevances $\text{rel}_q(d)$ of all items $d \in d^q$ are known. However, it is expensive to assess the relevance of all items exhaustively.

An alternate source of relevance labels is implicit feedback, as it is available in virtually unlimited quantities. However, the relevance labels rel_q are typically not fully revealed by implicit feedback. Instead, implicit feedback only provides partial information about the relevances for a subset of the candidates. We denote this partial feedback as c_q, where $c_q(d) = 1$ indicates positive feedback (e.g., click) for item d and $c_q(d) = 0$ indicates the absence of positive feedback (e.g., no click). We call this the partial-information setting [30].

The goal of LTR is learning a ranking policy π from a class of ranking policies Π using the available feedback data for the training queries in Q. Unlike most other works on LTR, we consider stochastic ranking policies, where $\pi(r|q)$ is a distribution over the rankings r of the candidate set. As will become clear later, stochastic ranking policies have the advantage of providing more fine-grained control of exposure and enable gradient-based optimization. Note that deterministic ranking policies are a special case of stochastic ranking policies, where all probability mass lies on a single ranking.

In conventional LTR algorithms [11–13, 27], the key objective is to learn a policy π that maximizes the utility $U(\pi)$ to the users

$$U(\pi) = \mathbb{E}_{q\sim P(q)}\left[U(\pi|q)\right] = \mathbb{E}_{q\sim P(q)}\,\mathbb{E}_{r\sim\pi(r|q)}\left[\Delta(r, \text{rel}_q)\right]. \quad (1)$$

The utility of a ranking r for a query q can be captured by any ranking metric Δ, like Discounted Cumulative Gain (DCG) [26]. However, utility optimization by itself does not ensure fairness [42]. We, therefore, include an additional constraint that addresses intrinsic fairness by enforcing an application-dependent allocation of exposure based on merit. For simplicity of notation, consider the case of two groups G_i and G_j. To ensure that exposure is allocated fairly between G_i and G_j, we measure unfairness via a disparity measure $D_{ij}(\pi)$, which will be defined in Section 3.3. A perfectly fair ranking policy has disparity zero. However, more generally, we may want to restrict $[D_{ij}(\pi)]^2$ to be no more than some threshold δ. Combining the fairness constraint with the conventional goal of optimizing utility in LTR, we define our objective as

$$\pi^* = \underset{\pi\in\Pi}{\text{argmax}}\, U(\pi) \text{ s.t. } [D_{ij}(\pi)]^2 \leq \delta.$$

Directly optimizing this objective is not possible for at least two reasons. First, we do not know the query distribution $P(q)$ to compute the expectation in Equation (1), but we merely have access to the sample $\boldsymbol{Q}$. Second, even computing the utility $U(\pi|q)$ for an individual query is problematic, since the relevances rel_q are only partially revealed by the implicit feedback c_q. The same is true for the fairness divergence $D_{ij}(\pi)$ as defined below, since it also depends on relevance. We thus need to replace $U(\pi)$ and $D_{ij}(\pi)$ with estimators $\widehat{U}(\pi|\boldsymbol{Q})$ and $\widehat{D}_{ij}(\pi|\boldsymbol{Q})$ based on the query sample $\boldsymbol{Q}$ and the implicit feedback c_q, leading to the following constrained Empirical Risk Minimization (ERM) objective.

$$\widehat{\pi} = \underset{\pi\in\Pi}{\text{argmax}}\, \widehat{U}(\pi|\boldsymbol{Q}) \text{ s.t. } \left[\widehat{D}_{ij}(\pi|\boldsymbol{Q})\right]^2 \leq \delta. \quad (2)$$

This leads to three challenges that we address in the following. First, in Section 3.2, we show how to design $\widehat{U}(\pi|\boldsymbol{Q})$ to get unbiased utility estimates even if we do not have access to true relevance labels rel_q but only have access to biased implicit feedback c_q. Second, in Section 3.3, we show how to define estimators of the fairness disparity $\widehat{D}_{ij}(\pi)$ that ensure merit-based exposure allocation while also being unbiased even with only the implicit feedback c_q. Finally, in Section 4, we show how to efficiently solve the resulting training problem from Equation (2).

3.2 Unbiased Utility Estimator

We begin by defining the ranking metric that we use to measure the utility of ranking r for query q. We consider the class of additive ranking metrics, which can be expressed as

$$\Delta(r, \text{rel}_q) = \sum_{d\in d^q} f\left(r(d)\right)\cdot \text{rel}_q(d),$$

where $r(d)$ denotes the rank of item d in ranking r and $f()$ can be any weighting function that depends on $r(d)$. For example, for the DCG metric, we can set $f(r(d)) = 1/\log_2(1 + r(d))$, while for the Average Rank metric, we set $f(r(d)) = -r(d)$. For simplicity, we assume binary relevances, i.e., $\text{rel}_q(d) \in \{0, 1\}$.

In the partial-information setting, the relevances rel_q are not directly available. Furthermore, naively treating c_q as a proxy for rel_q can suffer from presentation bias. For example, relevant items ranked at top positions in the presented ranking r_q are more likely to be clicked by users than those ranked at lower positions, confounding the relevance signal we would like to train on. As a result, the learned policy will be skewed towards items already ranked at top positions by the logging policy, leading to rich-get-richer dynamics.

To overcome the presentation bias of implicit feedback, following [30], we introduce the binary random variable $o_q(d)$ indicating whether the item d is examined by the user. Based on this, we can model the distribution of $o_q(d)$, especially the "propensity" $p(o_q(d) = 1|r_q)$, of observing $\text{rel}_q(d)$ given that ranking r_q was presented when the implicit feedback was logged. With knowledge of the propensity, we can use Inverse Propensity Score (IPS) weighting to arrive at the following estimator $\widehat{\Delta}$ for the ranking metric Δ [30]

$$\widehat{\Delta}(r, c_q) = \sum_{d:c_q(d)=1} \frac{f\left(r(d)\right)}{p(o_q(d) = 1|r_q)}.$$

The estimator is unbiased if all propensities are bounded away from zero [30]. Furthermore, we can define an estimator of the utility as

$$\widehat{U}(\pi|\boldsymbol{Q}) = \frac{1}{|\boldsymbol{Q}|}\sum_{q\in\boldsymbol{Q}}\widehat{U}(\pi|q) = \frac{1}{|\boldsymbol{Q}|}\sum_{q\in\boldsymbol{Q}}\mathbb{E}_{r\sim\pi(r|q)}[\widehat{\Delta}(r,c_q)].$$

It is easy to verify that this estimator inherits unbiasedness from the unbiasedness of $\widehat{\Delta}$. Furthermore, the estimator is consistent under the same condition on the propensities, thus implying that $\widehat{U}(\pi|\boldsymbol{Q})$ will converge to $U(\pi)$ as $|\boldsymbol{Q}|$ increases.

There are multiple choices for modeling the propensity $p(o_q(d) = 1|r_q)$ [20, 30]. The most common and simplest one is the position-based examination model, where $p(o_q(d) = 1|r_q)$ depends on only the position $r_q(d)$ of d in the ranking r_q presented during logging,

$$p(o_q(d) = 1|r_q) = v_{r_q(d)}.$$

Here v_k denotes the examination probability at position k, also referred to as position bias. A click is observed ($c_q(d) = 1$) whenever an item is examined ($o_q(d) = 1$) and relevant ($\mathrm{rel}_q(d) = 1$), with the addition of noise as discussed in Section 3.5. More elaborate models also take contextual information beyond rank into account [20]. Either model can be estimated with swap experiments [30] or intervention harvesting [4, 20].

This resolves the first extrinsic source of unfairness, namely that the ranking system uses a biased $\widehat{U}(\pi|\boldsymbol{Q})$ that is corrupted by position bias. However, an unbiased $\widehat{U}(\pi|\boldsymbol{Q})$ alone does not guarantee the fairness of exposure, as discussed in the next section.

3.3 Unbiased Fairness Constraints

The work in [42] has shown that an LTR algorithm maximizing utility $U(\pi)$ can be unfair even if $U(\pi)$ is perfectly known. Therefore we want to enforce additional criteria of how exposure is allocated based on merit to counteract such intrinsic sources of unfairness. In our training problem from Equation (2), this is implemented through the disparity measure $\widehat{D}_{ij}(\pi|\boldsymbol{Q})$ in the constraint, which we now define formally.

As a first step, we define the exposure of an item d in ranking r as the probability that a user accessing the ranking will examine the item. This is identical to the examination probability $p(o_q(d) = 1|r)$ defined in Section 3.2, but now this model is applied to not only the logged ranking r_q but all rankings. The exposure of d under a stochastic ranking policy π for a query q, denoted as $\mathrm{Exp}_q(d|\pi)$, is the expected exposure over all the possible rankings

$$\mathrm{Exp}_q(d|\pi) = \mathbb{E}_{r\sim\pi(r|q)}\left[p(o_q(d) = 1|r)\right].$$

Furthermore, the exposure of group G_i is the aggregate of the exposure of the group members

$$\mathrm{Exp}_q(G_i|\pi) = \sum_{d\in G_i^q}\mathrm{Exp}_q(d|\pi), \tag{3}$$

where $G_i^q = G_i \cap d^q$. Similarly, we define the relevance of group G_i for query q as

$$\mathrm{rel}_q(G_i) = \sum_{d\in G_i^q}\mathrm{rel}_q(d). \tag{4}$$

With these definitions in hand, we define our fairness disparity of policy π as

$$D_{ij}(\pi) = \mathbb{E}_{q\sim Q}[D_{ij}(\pi|q)], \tag{5}$$

where $D_{ij}(\pi|q)$ measures the disparate exposure of G_i and G_j based on their merit (i.e., relevances) for query q as

$$D_{ij}(\pi|q) = \mathrm{rel}_q(G_j)\,\mathrm{Exp}_q(G_i|\pi) - \mathrm{rel}_q(G_i)\,\mathrm{Exp}_q(G_j|\pi). \tag{6}$$

Note that the disparity is zero when the following proportionality between merit and exposure from [42, 43]

$$\frac{\mathrm{Exp}_q(G_i|\pi)}{\mathrm{rel}_q(G_i)} = \frac{\mathrm{Exp}_q(G_j|\pi)}{\mathrm{rel}_q(G_j)}$$

is fulfilled for all queries q. The constraint says that exposure should be allocated to each group proportional to the group's merit, although other merit-based allocation schemes can be implemented as well [42].

However, fulfilling the constraint for each query is only a sufficient but not necessary condition for the disparity $D_{ij}(\pi)$ to be zero. As we will discuss in more detail in Section 3.4, the disparity corresponds to an amortized version of fairness of exposure similar to [43].

While $\mathrm{Exp}_q(G_i|\pi)$ is an expectation that can be computed, we do have to estimate $D_{ij}(\pi)$ with respect to the unknown query distribution and the unknown relevances. We therefore take the empirical counterpart of $D_{ij}(\pi)$ as

$$\widehat{D}_{ij}(\pi|\boldsymbol{Q}) = \frac{1}{|\boldsymbol{Q}|}\sum_{q\in\boldsymbol{Q}}\widehat{D}_{ij}(\pi|q). \tag{7}$$

Furthermore, in the partial-information setting, we can get an unbiased estimate of $\mathrm{rel}_q(G_i)$ using the IPS estimator

$$\widehat{\mathrm{rel}}_q(G_i) = \sum_{d\in G_i^q}\frac{c_q(d)}{p(o_q(d) = 1|r_q)}. \tag{8}$$

This means that $\mathrm{rel}_q(G_i)$ in Equation (6) is replaced with $\widehat{\mathrm{rel}}_q(G_i)$ to arrive at the following empirical disparity measure

$$\widehat{D}_{ij}(\pi|q) = \widehat{\mathrm{rel}}_q(G_j)\,\mathrm{Exp}_q(G_i|\pi) - \widehat{\mathrm{rel}}_q(G_i)\,\mathrm{Exp}_q(G_j|\pi). \tag{9}$$

Note that $\widehat{D}_{ij}(\pi|q)$ is unbiased since $\widehat{\mathrm{rel}}_q(G_i)$ is unbiased and the exposure terms are non-random constants,

$$\begin{aligned}
&\mathbb{E}_{o_q}\left[\widehat{D}_{ij}(\pi|q)\right]\\
=&\mathbb{E}_{o_q}\left[\widehat{\mathrm{rel}}_q(G_j)\,\mathrm{Exp}_q(G_i|\pi) - \widehat{\mathrm{rel}}_q(G_i)\,\mathrm{Exp}_q(G_j|\pi)\right]\\
=&\mathbb{E}_{o_q}[\widehat{\mathrm{rel}}_q(G_j)]\,\mathrm{Exp}_q(G_i|\pi) - \mathbb{E}_{o_q}[\widehat{\mathrm{rel}}_q(G_i)]\,\mathrm{Exp}_q(G_j|\pi)\\
=&\mathrm{rel}_q(G_j)\,\mathrm{Exp}_q(G_i|\pi) - \mathrm{rel}_q(G_i)\,\mathrm{Exp}_q(G_j|\pi)\\
=&D_{ij}(\pi|q).
\end{aligned}$$

Since $\widehat{D}_{ij}(\pi|q)$ is unbiased for each query q, the aggregate $\widehat{D}_{ij}(\pi|\boldsymbol{Q})$ is also unbiased for $D_{ij}(\pi)$,

$$\mathbb{E}_q\,\mathbb{E}_{o_q}[\widehat{D}_{ij}(\pi|\boldsymbol{Q})] = \mathbb{E}_q[D_{ij}(\pi|q)] = D_{ij}(\pi).$$

Furthermore, through the use of Hoeffding bounds, it is possible to show that $\widehat{D}_{ij}(\pi|\boldsymbol{Q})$ converges to the true disparity $D_{ij}(\pi)$ as $|\boldsymbol{Q}|$ increases.

3.4 Relation to other Disparity Measures

In this section, we further investigate the relationship between our disparity measure from Equation (6) and the Disparate Treatment constraint as proposed by Singh and Joachims [42]. We show that under some assumptions, optimizing our disparity measure can be similar to optimizing an amortized version of the Disparate Treatment constraint in [42]. However, when such assumptions do not hold, either disparity measure can be chosen. However, our measure of disparity provides additional advantages.

The Disparate Treatment constraint as mentioned in [42] states

$$\forall q : \frac{\text{Exp}_q(G_i|\pi)}{\text{rel}_q(G_i)} = \frac{\text{Exp}_q(G_j|\pi)}{\text{rel}_q(G_j)}.$$

This constraint expresses that for a given query, the exposure of each group should be proportional to its relevance. However, this constraint is difficult to implement with implicit feedback, since we need an unbiased estimator that eliminates the effect of presentation bias on relevances in this fairness constraint. Specifically, to estimate $1/\text{rel}_q(G_i)$ using c_q, we have to know the joint distribution of $o_q(d)$ for all the items in G_i^q, which is difficult to model. To avoid this obstacle, we transform the Disparate Treatment constraint by multiplying both sides with $\text{rel}_q(G_i)\ \text{rel}_q(G_j)$

$$\begin{aligned}&\text{Exp}_q(G_i|\pi)/\text{rel}_q(G_i) = \text{Exp}_q(G_j|\pi)/\text{rel}_q(G_j)\\ \Leftrightarrow\ &\text{rel}_q(G_j)\,\text{Exp}_q(G_i|\pi) = \text{rel}_q(G_i)\,\text{Exp}_q(G_j|\pi).\end{aligned}$$

This leads to our measure of disparity in Equation (6).

An alternate path towards a closely related disparity measure using implicit feedback is the following. We replace the query-specific exposure and relevances with an amortized notion of exposure and relevances [9] over the query distribution

$$\frac{\mathbb{E}_q[\text{Exp}_q(G_i|\pi)]}{\mathbb{E}_q\left[\text{rel}_q(G_i)\right]} = \frac{\mathbb{E}_q[\text{Exp}_q(G_j|\pi)]}{\mathbb{E}_q\left[\text{rel}_q(G_j)\right]}.$$

This constraint expresses that for all the queries, the amortized exposure of each group should be proportional to its amortized relevance. We will show that the disparity

$$D'_{ij}(\pi) = \frac{\mathbb{E}_q[\text{Exp}_q(G_i|\pi)]}{\mathbb{E}_q[\text{rel}_q(G_i)]} - \frac{\mathbb{E}_q[\text{Exp}_q(G_j|\pi)]}{\mathbb{E}_q[\text{rel}_q(G_j)]} \tag{10}$$

is equivalent to disparity measure $D_{ij}(\pi)$ defined in the previous section under two assumptions. First, assume that the total exposure of both groups is a constant $\text{Exp}_q(G_i) + \text{Exp}_q(G_j) = C_E$. In practice, the top items receive most of the users' attention, so the total exposure for each query is relatively stable even if the size of the candidate set varies. Secondly, assume that the covariance of the exposure of group G_j and the total number of relevant items in both groups, $\text{cov}[\text{Exp}(G_j|\pi), \text{rel}_q(G_i) + \text{rel}_q(G_j)]$, is zero. One sufficient condition for this assumption to be approximately satisfied is when the numbers of relevant items do not vary much between queries. Under these two assumptions, the disparity measures $D_{ij}(\pi)$ and $D'_{ij}(\pi)$ are equivalent up to a constant coefficient

$$D_{ij}(\pi) = \mathbb{E}_q[\text{rel}_q(G_i)]\,\mathbb{E}_q[\text{rel}_q(G_j)]D'_{ij}(\pi).$$

This can be shown by transforming the disparity measure $D'_{ij}(\pi)$ as follows,

$$\begin{aligned}&\frac{\mathbb{E}_q\left[\text{Exp}_q(G_i|\pi)\right]}{\mathbb{E}_q[\text{rel}_q(G_i)]} - \frac{\mathbb{E}_q[\text{Exp}_q(G_j|\pi)]}{\mathbb{E}_q[\text{rel}_q(G_j)]}\\ =&\frac{\mathbb{E}_q[\text{Exp}_q(G_i|\pi)]\,\mathbb{E}_q[\text{rel}_q(G_j)] - \mathbb{E}_q[\text{Exp}_q(G_j|\pi)]\,\mathbb{E}_q[\text{rel}_q(G_i)]}{\mathbb{E}_q[\text{rel}_q(G_i)]\,\mathbb{E}_q[\text{rel}_q(G_j)]}\\ =&\frac{D_{ij}(\pi) + \text{cov}[\text{Exp}_q(G_j|\pi), \text{rel}_q(G_i)] - \text{cov}[\text{Exp}_q(G_i|\pi), \text{rel}_q(G_j)]}{\mathbb{E}_q[\text{rel}_q(G_i)]\,\mathbb{E}_q[\text{rel}_q(G_j)]}.\end{aligned}$$

The difference between the covariance terms is zero,

$$\begin{aligned}&\text{cov}[\text{Exp}_q(G_j|\pi), \text{rel}_q(G_i)] - \text{cov}[\text{Exp}_q(G_i|\pi), \text{rel}_q(G_j)]\\ =&\,\text{cov}[\text{Exp}_q(G_j|\pi), \text{rel}_q(G_i)] - \text{cov}[C_E - \text{Exp}_q(G_j|\pi), \text{rel}_q(G_j)]\\ =&\,\text{cov}[\text{Exp}(G_j|\pi), \text{rel}_q(G_i) + \text{rel}_q(G_j)] = 0.\end{aligned}$$

Therefore, we have $D_{ij}(\pi) = \mathbb{E}_q[\text{rel}_q(G_i)]\,\mathbb{E}_q[\text{rel}_q(G_j)]D'_{ij}(\pi)$.

Note that we can also get a consistent estimator of $D'_{ij}(\pi)$ by using the IPS estimator of relevances and replacing the expectations of exposure and relevance with the average over the dataset

$$\widehat{D}'_{ij}(\pi|\boldsymbol{Q}) = \frac{\sum_{q\in\boldsymbol{Q}} \text{Exp}_q(G_i|\pi)}{\sum_{q\in\boldsymbol{Q}} \widehat{\text{rel}}_q(G_i)} - \frac{\sum_{q\in\boldsymbol{Q}} \text{Exp}_q(G_j|\pi)}{\sum_{q\in\boldsymbol{Q}} \widehat{\text{rel}}_q(G_j)}.$$

As N increases, $\widehat{D}'_{ij}(\pi|\boldsymbol{Q})$ will converge to $D'_{ij}(\pi)$. However, the estimator is not unbiased, which means $\mathbb{E}[\widehat{D}'_{ij}(\pi|\boldsymbol{Q})] \neq D'_{ij}(\pi)$. This bias might increase the error of the estimator. We therefore prefer the disparity $D_{ij}(\pi)$ with its unbiased estimator $\widehat{D}_{ij}(\pi)$, as defined in Equations (7) and (9).

3.5 Incorporating Click Noise

Up to now, we assumed that feedback c_q is partial but that presence of feedback (i.e., $c_q(d) = 1$) reveals the relevance label $\text{rel}_q(d)$ in a noise-free way – precisely that $c_q(d) = 1$ if and only if $\text{rel}_q(d) = 1$ and $o_q(d) = 1$. In practice, the user might make mistakes when examining the relevances of items, and we now define the following noise model. With $1 \geq \epsilon_+ > \epsilon_- \geq 0$,

$$\begin{aligned}P\left(c_q(d) = 1|\text{rel}_i = 1, o_q(d) = 1\right) &= \epsilon_+,\\ P\left(c_q(d) = 1|\text{rel}_i = 0, o_q(d) = 1\right) &= \epsilon_-.\end{aligned}$$

If $\epsilon_+ < 1$, users might ignore relevant items even after examinations. If $\epsilon_- > 0$, users might give false positive feedback to irrelevant items after examinations. Fortunately, the IPS estimator of utility is order-preserving for this type of click noise [30], namely

$$\mathbb{E}_q\left[\widehat{U}(\pi_1|\boldsymbol{Q})\right] > \mathbb{E}_q\left[\widehat{U}(\pi_2|\boldsymbol{Q})\right] \Leftrightarrow U(\pi_1) > U(\pi_2|q).$$

Therefore, given enough data, the click noise does not affect the ability to find the ranking policy with optimal utility.

By contrast, this property does not hold for the disparity estimator. Specifically, we can observe that

$$\begin{aligned}
&\mathbb{E}_{o_q}\left[\widehat{\text{rel}}_q(G_i)\,\text{Exp}_q(G_j|\pi)\right]\\
=&\,\mathbb{E}_{o_q}\,\mathbb{E}_{c_q|o_q}\left[\sum_{d\in G_i^q}(c_q(d)/p)\,\text{Exp}_q(G_j|\pi)\right]\\
=&\sum_{d\in G_i^q}\left[\epsilon_+\text{rel}_q(d)+\epsilon_-(1-\text{rel}_q(d))\right]\text{Exp}_q(G_j|\pi)\\
=&(\epsilon_+-\epsilon_-)\text{rel}_q(G_i)\text{Exp}_q(G_j|\pi)+\epsilon_-|G_i^q|\text{Exp}_q(G_j|\pi),
\end{aligned}$$

where p stands for $p(o_q(d)=1|r_q)$. The expectation of the disparity estimator with click noise is

$$\begin{aligned}
&\mathbb{E}_{o_q}\left[\widehat{D}_{ij}(\pi|q)\right]\\
=&(\epsilon_+-\epsilon_-)\left(\text{rel}_q(G_j)\text{Exp}_q(G_i|\pi)-\text{rel}_q(G_i)\text{Exp}_q(G_j|\pi)\right)\\
&+\epsilon_-\left(|G_j^q|\text{Exp}_q(G_i|\pi)-|G_i^q|\text{Exp}_q(G_j|\pi)\right)\\
=&(\epsilon_+-\epsilon_-)D_{ij}(\pi|q)+\epsilon_-\left(|G_j^q|\text{Exp}_q(G_i|\pi)-|G_i^q|\text{Exp}_q(G_j|\pi)\right).
\end{aligned}$$

The additional term is dependent on the policy π. Therefore, the disparity estimator is not order-preserving,

$$\begin{aligned}
\mathbb{E}_q\Big[\widehat{D}_{ij}(\pi_1|\boldsymbol{Q})-\widehat{D}_{ij}(\pi_2|\boldsymbol{Q})\Big]=(\epsilon_+-\epsilon_-)\,\mathbb{E}_q\big[D_{ij}(\pi_1|\boldsymbol{Q})-D_{ij}(\pi_2|\boldsymbol{Q})\big]\\
+\epsilon_-\,\mathbb{E}_q\Big[|G_j^q|\delta\,\text{Exp}_q(G_i)-|G_i^q|\delta\,\text{Exp}_q(G_j)\Big],
\end{aligned}$$

where $\delta\,\text{Exp}(G_i^q)$ stands for $\text{Exp}_q(G_i|\pi_1)-\text{Exp}_q(G_i|\pi_2)$. This implies that when $\epsilon_-=0$, our IPS estimator for group disparity is consistent for finding an optimal ranking policy regardless of the value of ϵ_+. However, when $\epsilon_->0$, the disparity estimator is not order-preserving, which means that there can exist two policies π_1 and π_2 such that $\mathbb{E}\left[D_{ij}(\pi_1|\boldsymbol{Q})\right]>\mathbb{E}\left[D_{ij}(\pi_2|\boldsymbol{Q})\right]$ but $\widehat{D}_{ij}(\pi_1)<\widehat{D}_{ij}(\pi_2)$.

Given the level of negative noise ϵ_-, we can correct the IPS estimator for group disparity as

$$\begin{aligned}
\widehat{D}_{ij}(\pi|q,\epsilon_-)=&\,\widehat{\text{rel}}_q(G_j)\text{Exp}_q(G_i|\pi)-\widehat{\text{rel}}_q(G_i)\text{Exp}_q(G_j|\pi)\\
&-\epsilon_-\left(|G_j^q|\text{Exp}_q(G_i|\pi)-|G_i^q|\text{Exp}_q(G_j|\pi)\right).
\end{aligned}\tag{11}$$

Various methods have been proposed for estimating the position bias vector v for search engines [4, 30, 44]. To get the probability ϵ_- for false-positive noise, we can use a simple intervention similar to [30]. While identifying relevant items is difficult, identifying irrelevant items is easier using a simple criteria (e.g., covering no terms in the query). Therefore, we can insert an item d that is known to be irrelevant at position k of the ranking result. The expected clickthrough rate for the item is

$$p(c_q(d)=1|\text{rel}_q(d)=0)=v_k\cdot\epsilon_-.$$

This means that given the position bias and the clickthrough rate estimated from the intervention data, we can compute the noise level ϵ_-. Note that the intervention only needs to be performed on a small set of users and only affects the quality of one position.

4 POLICY-GRADIENT ALGORITHM FOR FAIR LTR

In the previous section, we defined a general framework for learning ranking policies from biased and noisy feedback under amortized fairness of exposure constraints. However, we still need an efficient algorithm for searching the constrained policy space for the solution of the training problem in Equation (2). To this effect, we first define a stochastic ranking policy space based on the Plackett-Luce ranking model as in [43] and then present a policy-gradient algorithm that optimizes the training objective.

4.1 Plackett-Luce Ranking Model

We define a stochastic ranking policy space Π based on the Plackett-Luce model [31, 36]. Specifically, each ranking policy $\pi\in\Pi$ is defined by a scoring function h_θ. h_θ can be any differentiable machine learning model with parameters θ. h_θ takes the feature vectors $x_q(d)$ of all items d for the current query q as input and outputs a vector of scores $h_\theta(x_q)=(h_\theta(x_q(d_1)),h_\theta(x_q(d_2)),\cdots,h_\theta(x_q(d_{n_q})))$. Based on this score vector, the probability $\pi_\theta(r|q)$ of a ranking $r=<d_1,d_2,\cdots,d_{n_q}>$ is defined as the product of softmax distributions

$$\pi_\theta(r|q)=\prod_{i=1}^{n_q}\frac{\exp(h_\theta(x_q(d_i)))}{\sum_{j=i}^{n_q}\exp(h_\theta(x_q(d_j)))}.\tag{12}$$

Sampling rankings from $\pi_\theta(r|q)$ is quite straightforward and efficient since we use Monte-Carlo estimates over this distribution of rankings. It can be implemented as sampling from the distribution softmax$(h_\theta(x_q))$ without replacement and ranking the items according to the order in which they are drawn.

4.2 Policy Gradient Training Algorithm

Optimizing the objective in Equation (2) is a constrained optimization problem. We use a Lagrange multiplier to solve the problem via

$$\hat{\pi}=\operatorname*{argmax}_\pi\min_{\lambda\geq0}\widehat{U}(\pi|\boldsymbol{Q})-\lambda\Big(\left[\widehat{D}_{ij}(\pi|\boldsymbol{Q})\right]^2-\delta\Big).$$

Instead of solving the minimization problem w.r.t. λ, we search a specific range of $\lambda\in\{\lambda_1,\cdots,\lambda_k\}$. For each λ, we need to solve

$$\hat{\pi}_\lambda=\operatorname*{argmax}_\pi\widehat{U}(\pi|\boldsymbol{Q})-\lambda\left[\widehat{D}_{ij}(\pi|\boldsymbol{Q})\right]^2.\tag{13}$$

Afterwards, we can compute the corresponding $\delta_\lambda=\left[\widehat{D}_{ij}(\hat{\pi}_\lambda|\boldsymbol{Q})\right]^2$ for which the constraint holds. We can then pick the optimal $\hat{\pi}_\lambda$ that satisfies $\delta_\lambda\leq\delta$ and provides maximal utility $\sum_q\widehat{U}(\hat{\pi}_\lambda|q)$.

It remains to find an efficient algorithm for solving the now unconstrained optimization problem in Equation (13). We use stochastic gradient descent (SGD) to iteratively update the parameters of the ranking policy. However, both $\widehat{U}(\pi|q)$ and $\widehat{D}_{ij}(\pi|q)$ are expectations over rankings, and it is intractable to compute these expectations over the exponential space of rankings. Following [43], we use sampling via the log-derivative trick of the REINFORCE algorithm [45] to compute the gradient of $\widehat{U}$

$$\begin{aligned}
\nabla_\theta\sum_q\widehat{U}(\pi_\theta|q)=&\nabla_\theta\sum_q\mathbb{E}_{r\sim\pi(r|q)}\left[\widehat{\Delta}(r,c^q)\right]\\
=&\sum_q\mathbb{E}_{r\sim\pi(r|q)}[\nabla_\theta\log\pi_\theta(r|q)\widehat{\Delta}(r,c^q)].
\end{aligned}\tag{14}$$

We also need the gradient of the squared fairness disparity. While the square of the disparity makes this more complex, the following transformation provides a quantity that again can be estimated via Monte-Carlo sampling,

$$\begin{aligned}&\nabla_\theta \left[\widehat{D}_{ij}(\pi_\theta|\boldsymbol{Q})\right]^2 \\ =&\frac{2}{|\boldsymbol{Q}|}\widehat{D}_{ij}(\pi_\theta|\boldsymbol{Q})\nabla_\theta\Big[\sum_{q\in\boldsymbol{Q}}\mathbb{E}_{r\sim\pi_\theta(r|q)}\,\widehat{\text{diff}}_{ij}(r|q)\Big] \\ =&\frac{2}{|\boldsymbol{Q}|}\widehat{D}_{ij}(\pi_\theta|\boldsymbol{Q})\sum_{q\in\boldsymbol{Q}}\mathbb{E}_{r\sim\pi_\theta(r|q)}\left[\nabla_\theta\log\pi_\theta(r|q)\widehat{\text{diff}}_{ij}(r|q)\right].\end{aligned} \tag{15}$$

Note that $\widehat{\text{diff}}_{ij}(r|q) = \widehat{M}^q_{G_j}\text{Exp}_q(G_i|r) - \widehat{M}^q_{G_i}\text{Exp}_q(G_j|r)$. If we apply stochastic gradient descent (SGD) for optimization, we have to compute $\widehat{D}_{ij}(\pi_{\theta_t}|\boldsymbol{Q})$ on the whole dataset at each step t, which is quite expensive. Another option is to use the stochastic gradient estimated by sampling $q_1, q_2 \sim \boldsymbol{Q}$ instead of iterating on the queries in the dataset,

$$\nabla_\theta\widehat{D}_{ij}(\pi|q_1,q_2) = 2\widehat{D}_{ij}(\pi_\theta|q_1)\,\mathbb{E}_{r\sim\pi_\theta}\left[\nabla_\theta\log\pi_\theta(r|q_2)\widehat{\text{diff}}_{ij}(r|q_2)\right].$$

This gradient estimator is unbiased as long as q_1 and q_2 are independently sampled from $\boldsymbol{Q}$. However, it can suffer from high variance since it is a product. As a trade-off between variance and bias, we use the running average $\frac{1}{n}\sum_{\tau=0}^{n-1}\widehat{D}_{ij}(\pi_{\theta_{t-\tau}}|q_{t-\tau})$ as an approximation of $\sum_q \widehat{D}_{ij}(\pi_{\theta_t}|q)/|\boldsymbol{Q}|$ in Equation (15). This is biased since the disparities are computed on previous parameters, but it can reduce the variance of sampling from the query set. In practice, we find this estimator to be very effective.

The expectations over rankings in Equations (14) and (15) are approximated via Monte-Carlo sampling from the policy $\pi_\theta(r|q)$. Following [43], we subtract a baseline term from the reward [45] to act as a control variate for variance reduction. The baseline is the average reward of the Monte-Carlo samples.

While optimizing over stochastic policies, entropy regularization is used as a method for encouraging exploration as to avoid premature convergence to suboptimal deterministic policies [32]. We therefore add the product of entropy of the probability distribution and a regularization coefficient γ to the objective. We initialize γ with a large value and reduce it when the validation metric has stopped improving.

5 EMPIRICAL EVALUATION

We conducted experiments on synthetic click data derived from the Microsoft Learning to Rank Dataset (Fold1) [37] and the German Credit Dataset [18]. This allows us to control the experimental conditions and test multiple data distributions to evaluate robustness.

The Microsoft LTR Dataset contains a large number of queries from Bing with manually-judged relevance labels. We adopt the train, validation, and test split provided with the dataset. We binarize relevances in the same way as [30], by assigning $\text{rel}_q(d) = 1$ to all the items that were judged as 3 or 4 and $\text{rel}_q(d) = 0$ to judgments 0, 1, and 2. After this step, the dataset becomes extremely sparse, with only about 2.5% relevant items per query. To better compare different methods and amplify differences, we remove queries with less than 20 candidates. For the remaining queries, we sample 20 candidate items with at most 3 relevant items for each query. Since the dataset does not come with any designated groups, we use the QualityScore (feature id 133) as the group attribute, dividing items into two groups with the 40th percentile as the threshold.

The German Credit Dataset contains information about 1000 individuals, where each individual is represented by a feature vector and labeled as creditworthy or non-creditworthy. We adapt it to an LTR task following [43]. We randomly split the 1000 individuals into train, validation, and test sets with ratio 1:1:1. For each query, we sample 20 individuals from the corresponding set with ratio 9:1 for non-creditworthy individuals to creditworthy individuals respectively. To define groups, we use the binary feature indicating whether the purpose is radio/television (attribute id A43) as the group attribute.

We generate click data for training and validation from the full-information datasets following [30]. We first train a conventional Ranking SVM with 1 percent of the full-information training data as the logging policy. This logging policy is then used to generate the rankings for which click data is logged. The click data is generated by simulating the position-based examination model. We use a position bias that decays with the presented rank k of the item as $v_k = (1/k)^\eta$. When not stated otherwise, we use $\eta = 1$.

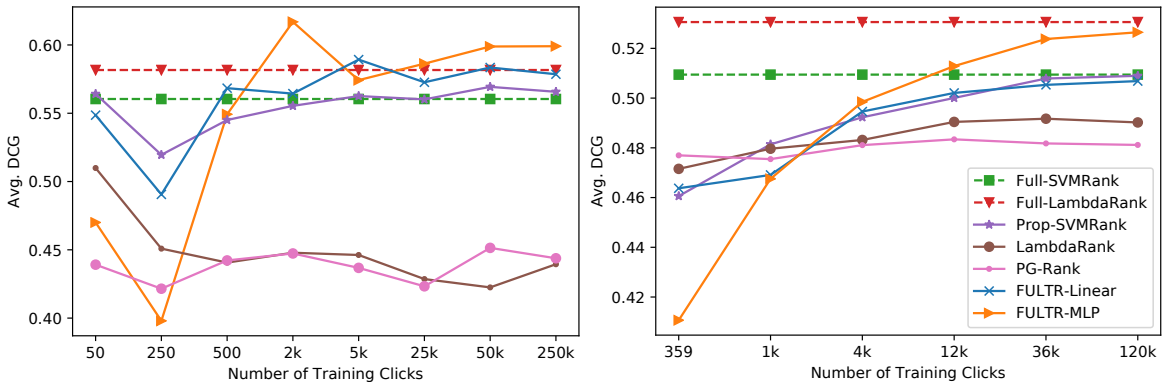
In all experiments, we select model hyperparameters via cross-validation on the click data generated from the validation set using the same criterion as in the respective training objective. The performance of the methods is reported on the full-information test set for which all relevance labels are known. Ranking utility and fairness are measured with Average DCG [26] and squared disparity $[D_{ij}(\pi)]^2$ respectively.

We train FULTR for two types of models: a linear model and a neural network (one hidden layer with ReLU activation). We use the SGD optimizer for the linear model and the Adam optimizer for the neural network. The learning rate is 0.001. We initialize the coefficient for entropy regularization as $\gamma = 1.0$ and reduce it by a factor of 3 each time the validation metric has stopped improving. We use a sample size of $S = 32$ for the Monte-Carlo estimates of the gradients. We add an L2 regularization term and cross-validate for the best regularization coefficient.

5.1 Can FULTR learn accurate ranking policies from biased feedback?

We begin the empirical evaluation by comparing FULTR without fairness constraints and conventional LTR methods when learning from partial click data. Here, we ignore fairness considerations but entirely focus on utility, just like in conventional LTR. For realism, we inject noise $\epsilon_- = 0.1$. We compare FULTR with the following conventional LTR methods, determining the hyperparameters based on the performance on the validation set:

- LambdaRank [11] is a nonlinear ranking model based on neural networks to optimize DCG. The hyperparameters are the L2 regularization coefficient and the learning rate.
- Propensity SVM-Rank [30] is an unbiased version of SVM-Rank [27] and utilizes the IPS estimator to correct position bias in implicit feedback. The hyperparameter is the regularization coefficient C.
- PG-Rank [43] is an analog of FULTR for the full-info setting. The hyperparameter is the L2 regularization coefficient.



(a) German Credit Dataset **(b) MSLR Dataset**

Figure 1: Ranking utility performance in terms of training clicks in the partial-info setting ($\eta = 1, \epsilon_- = 0.1$).

Note that the full-information methods LambdaRank and PG-Rank treat click signals naively as fully revealing the relevances, thus ignoring the position bias. However, we also add two skylines, Full-LambdaRank and Full-PG-Rank, which get to see all true relevance labels, not just the click signals. They represent the maximum performance we could hope to achieve with FULTR, which has access to only strictly less informative click feedback. For the purpose of evaluation in this task, we use the highest probability ranking of the candidate set for each query to compute the metrics over all the test set queries.

In Figure 1, we show the test-set performance of the conventional LTR methods compared to FULTR. With increasing amounts of click data on the x-axis, the neural-network version of FULTR approaches or exceeds the skyline performance of Full-LambdaRank on both datasets. FULTR with the linear model approaches the skyline performance of the linear Full-SVM-Rank on the MSLR dataset, but it outperforms the skyline performance of linear Full-SVM-Rank on the German Credit dataset. We conjecture that this is due to FULTR's ability to directly optimize utility, instead of optimizing a loose upper bound like in SVM-Rank.

Baseline methods that naively ignore position bias, namely LambdaRank and PG-Rank, cannot make effective use of the increased amount of click data. Their performance is well below FULTR once a sufficient amount of click data is available, and their learning curves are quite flat.

Overall, we conclude that FULTR is an LTR algorithm that achieves state-of-the-art ranking performance when learning from partial-information feedback. This implies that FULTR is a strong contender for use in practical applications, and it thus makes sense to further investigate how far it can also enforce fairness considerations.

5.2 Can FULTR learn fair ranking policies from biased feedback?

Next, we investigate FULTR's ability to enforce the merit-based fairness of exposure. As baselines for comparison, we also implemented the following methods:

- Group-blind is a version of FULTR without any fairness constraints but with the group attribute masked (i.e., fairness through unawareness).
- Fair-PG-Rank [43] is also a policy gradient method, but it naively treats click data as an unbiased relevance signal.
- Equity of Attention, as proposed by Biega et al. [9], is a post-processing method that optimizes amortized group disparity.

Since the post-processing method requires relevance estimates for all items, we train a regression model $f(x_q(d))$ on all query-item pairs in the training set using an unbiased IPS objective for relevance estimation proposed in [6],

$$\mathcal{L} = \sum_q \sum_{d \in d^q} \left([f(x_q(d))]^2 - \frac{2c_q(d)}{p} f(x_q(d)) + \frac{c_q(d)}{p} \right).$$

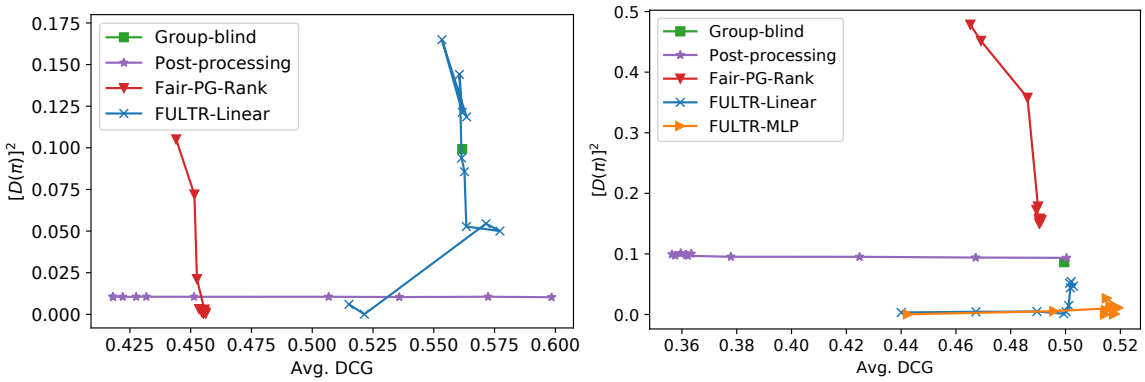
The resulting utility/disparity curves on the test set are shown in Figure 2. First, note that Group Blind training does not automatically lead to merit-based fairness of exposure.

Second, FULTR is able to effectively trade-off utility and fairness on both datasets as we vary the trade-off parameter λ in the objective. The pattern of the DCG/disparity curve is similar on both datasets – as λ increases, the disparity goes down with an associated drop in DCG as expected. Further increasing λ drives disparity close to zero. The neural ranking model learned by FULTR can achieve better utility with lower disparity compared with the linear model on the MSLR dataset. The neural network version, FULTR-MLP, is not evaluated on the German Credit dataset since the small scale of the dataset made it difficult to control over-fitting.

Third, Fair-PG-Rank does not behave in a predictable manner. As we vary its trade-off parameter λ, utility drops while the disparity increases on the MSLR dataset. We conjecture that this is due to two reasons. Firstly, Fair-PG-Rank optimizes disparity per query, instead of amortized disparity, and secondly, biased clicks lead to a biased utility objective and bias in the disparity measure of Fair-PG-Rank.

Table 1: Accumulated regression error for both groups.

Dataset	G_i	G_j
MSLR	-14.05%	-0.09%
German Credit	-15.86%	-19.47%



(a) German Credit Dataset ($N = 5K$) **(b) MSLR Dataset ($N = 12K$)**

Figure 2: DCG-Disparity curves for FULTR against baselines in the partial-info setting ($\eta = 1, \epsilon_- = 0$).

Fourth, the post-processing method cannot reduce amortized disparity on the MSLR dataset. To exclude that this failure is due to the slight difference between disparity definitions, we ran the post-processing method with ground-truth relevances. This method, although infeasible in practice, achieves disparities close to zero on MSLR. On the German Credit dataset, however, the post-processing method can approach zero disparity with high DCG. To explain the difference between the two datasets, we show the accumulated regression errors on both datasets in Table 1. The error is computed as $(\sum_q \sum_{d \in G_i^q} f(x_q(d)) - \text{rel}_q(G_i))/\sum_q \text{rel}_q(G_i)$. We observe that on the MSLR dataset, the regression model underestimates the relevance of G_i but estimates the relevance of G_j precisely, which means that the regression model is already unfair between groups despite its provably unbiased training objective. Naturally, the post-processing method cannot generate fair rankings from an unfair relevance model. On the German Credit dataset, the regression model underestimates the relevance of two groups uniformly, so the post-processing method can correct the ranking to be fair. Therefore, we conclude that the failure of the post-processing is due to its two-step nature, where the regression model is trained oblivious to fairness. In contrast, FULTR is trained end-to-end in one step, such that it can, for example, discount features that lead to unfair relevance estimates.

5.3 Can FULTR converge to the performance of training on the true relevance labels?

We now explore how ranking utility and fairness of FULTR converge as the learning algorithm is given additional click data. The DCG/Disparity curves with various amounts of training clicks are shown in Figure 3. We also show the policy learned by a straightforward adaptation of FULTR to the full-info setting, which serves as the skyline that has full knowledge of all relevance labels in the training set without position bias. With increasing amounts of click data, FULTR approaches the skyline performance of the policy learned on the full-info data. The policy learned on 120k partial-information examples is almost identical to that of the full-information policy. This demonstrates that an unbiased estimator of FULTR enables it to converge to the full-information policy given enough click data. Furthermore, we observe that with the maximal

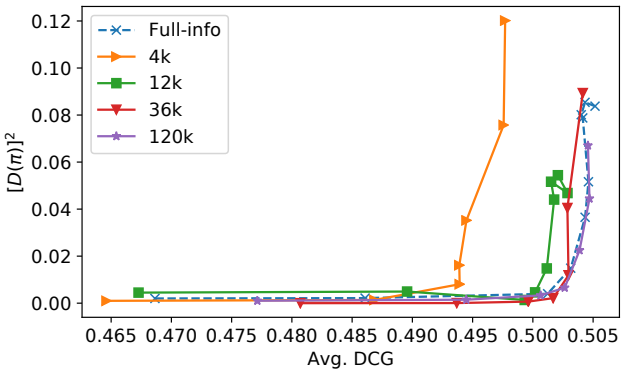


Figure 3: DCG-Disparity curves for FULTR in terms of training clicks on the partial MSLR dataset ($\eta = 1, \epsilon_- = 0$). The skyline is the policy learned in the full-info setting.

λ, we achieve disparity close to zero over a substantial range of click-data quantities. However, as expected, larger amounts of click data lead to higher utility as measured by DCG.

5.4 How important is unbiasedness for utility and fairness?

We now conduct an ablation study to understand the effect of IPS-weighting on utility and fairness. In particular, we compare the following variants of FULTR:

- No-IPS FULTR-Linear: Both utility and disparity estimators do not use IPS weighting and assume that implicit feedback reveals full-information relevance labels.
- Utility-IPS FULTR-Linear: The utility estimator is IPS-weighted, but the disparity estimator is unweighted.

The results are shown in Figure 5. With biased estimates of both utility and disparity, the *No-IPS* variant of FULTR achieves suboptimal utility and cannot reduce disparity to zero. The *Utility-IPS* variant can achieve utility similar to that of FULTR, but is fairness disparity remains higher even for large values of λ. This implies eliminating selection bias through IPS-weighting is essential for both utility and fairness.

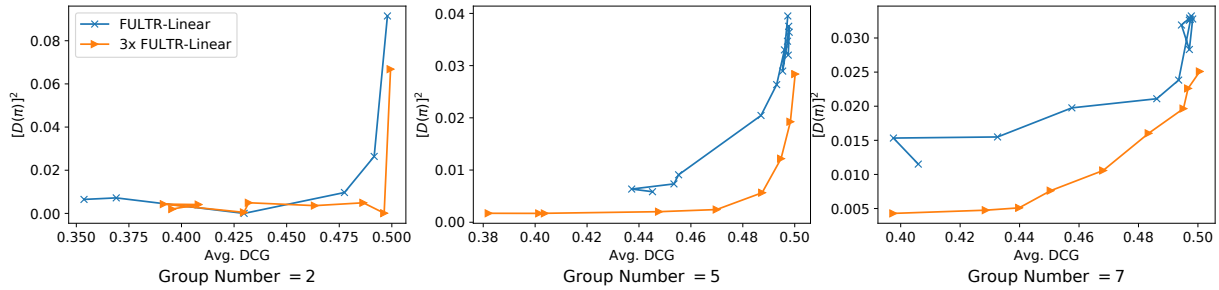


Figure 4: DCG-Disparity curves for different group numbers on the partial MSLR dataset ($N = 4K$ and $N = 12K$, $\eta = 1, \epsilon_- = 0$).

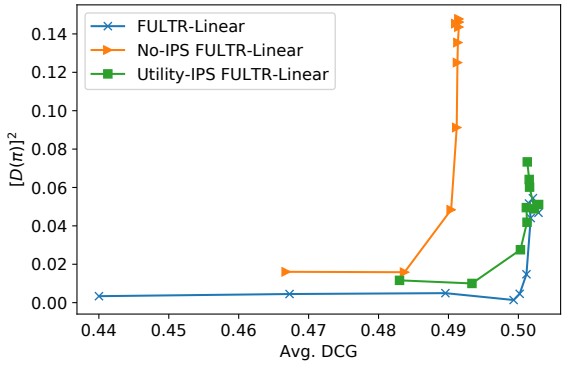


Figure 5: DCG-Disparity curves for FULTR and two variations with unweighted utility and disparity estimators on the partial MSLR dataset ($N = 12K, \eta = 1, \epsilon_- = 0$).

5.5 Can the disparity estimator adjust to click noise?

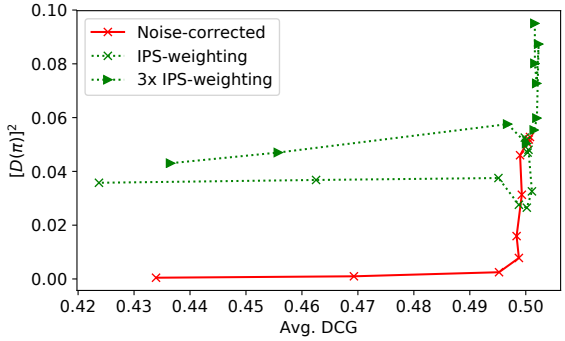


Figure 6: DCG-Disparity curves of noise-corrected estimators against pure IPS-weighting on the partial MSLR dataset with click noise ($N = 36K$ and $N = 120K$, $\eta = 1, \epsilon_- = 0.1$).

Next, we investigate the effectiveness of the noise-corrected disparity estimator in Equation (11). We use the MSLR dataset with click noise set to $\epsilon_- = 0.1$. Figure 6 shows that the noise-corrected IPS estimator of $D_{ij}(\pi)$ can reduce disparity effectively and achieve zero disparity, while the IPS estimator without noise correction cannot reduce disparity effectively. To verify that this is not due to the lack of data, we increase the amount of training data by a factor of three. However, more data alone does not help remedy the problem. This is consistent with the theoretical argument in Section 3.5, since more training data can only reduce the variance, while click noise leads to a bias in the disparity estimates.

5.6 Can FULTR ensure fairness between more than two groups?

So far, all experiments are conducted with only two groups. However, our method can also deal with multiple groups by modifying the objective in Equation (2) as

$$\pi^* = \operatorname*{argmax}_{\pi \in \Pi} \widehat{U}(\pi|\boldsymbol{Q}) \text{ s.t. } \sum_{G_i, G_j} \left[\widehat{D}_{ij}(\pi|\boldsymbol{Q})\right]^2 \leq \delta.$$

To validate the effectiveness of FULTR when dealing with multiple groups, we compare the utility-fairness trade-off for a varying number of groups in Figure 4. Following the previous setting, we use QualityScore as the group attribute and partition it into intervals with equal numbers of items. Figure 4 shows the performance for 2, 5, and 7 groups. We can observe that with a fixed number of training clicks, the disparity of FULTR increases as the number of groups increases. With 7 groups, FULTR still has a substantial disparity even for the largest value of λ. This is due to a lack of data for each group, as the number of groups is increasing while the amount of training data remains fixed. However, if we increase the number of training clicks, FULTR can again achieve disparity close to zero. This demonstrates that FULTR can deal with multiple groups, but with more number of groups, FULTR requires more training data.

6 CONCLUSION

We presented a framework, FULTR, for learning accurate and fair ranking policies from biased feedback that addresses both intrinsic and extrinsic sources of unfairness. Specifically, we introduced fairness-of-exposure constraints that can allocate amortized exposure to the different groups of items based on their amortized relevance. Furthermore, we proposed counterfactual estimators of the corresponding disparity measure and utility that remove the effects of position bias. Both estimators are shown to be unbiased, and we derived a policy gradient training algorithm that directly optimizes both utility and fairness. We also considered click noise and provided a noise-corrected estimator of disparity. Furthermore, we presented extensive empirical evidence that FULTR can effectively learn ranking policies under fairness constraints despite biased and noisy feedback.

REFERENCES

[1] Lada A Adamic and Bernardo A Huberman. 2000. Power-law distribution of the world wide web. *science* 287, 5461 (2000), 2115–2115.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453* (2018).

[3] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In *The World Wide Web Conference (2019)*. 4–14.

[4] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In *ACM International Conference on Web Search and Data Mining (2019)*. 474–482.

[5] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2018)*. 385–394.

[6] Anonymous. 2019. Controlling Fairness and Bias in Dynamic Ranking. *Under review as a conference submission* (2019).

[7] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409* (2017).

[8] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. *arXiv preprint arXiv:1903.00780* (2019).

[9] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2018)*. 405–414.

[10] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *International Conference on World Wide Web (2016)*. 531–541.

[11] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems (2006)*. 193–200.

[12] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *International Conference on Machine Learning (2005)*. 89–96.

[13] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (2007)*. 129–136.

[14] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. 2017. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840* (2017).

[15] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *International Conference on World Wide Web (2009)*. ACM, 1–10.

[16] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7, 3 (2015), 1–115.

[17] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *ACM International Conference on Web Search and Data Mining (2008)*. 87–94.

[18] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[19] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2008)*. 331–338.

[20] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-Dependent Examination-Bias Estimation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)*. 825–834.

[21] Fabrizio Germano, Vicenç Gómez, and Gaël Le Mens. 2019. The few-get-richer: a surprising consequence of popularity-based rankings?. In *The World Wide Web Conference (2019)*. ACM, 2764–2770.

[22] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)*. 2221–2231.

[23] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. 2018. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems (2018)*. 2600–2609.

[24] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *ACM International Conference on Web Search and Data Mining (2009)*. 124–131.

[25] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2018. A Novel Algorithm for Unbiased Learning to Rank. *arXiv preprint arXiv:1809.05818* (2018).

[26] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[27] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)*. 133–142.

[28] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems* 25, 2 (2007), 7.

[29] Thorsten Joachims, Laura A Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2005)*, Vol. 5. 154–161.

[30] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *ACM International Conference on Web Search and Data Mining (2017)*. 781–789.

[31] R. Duncan Luce. 1959. *Individual Choice Behavior: A Theoretical analysis*. Wiley.

[32] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning (2016)*. 1928–1937.

[33] Harikrishna Narasimhan. 2018. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics (2018)*. 1646–1654.

[34] Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Serena Wang. 2019. Pairwise Fairness for Ranking and Regression. *arXiv preprint arXiv:1906.05330* (2019).

[35] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Y Narahari. 2019. Achieving Fairness in the Stochastic Multi-armed Bandit Problem. *arXiv preprint arXiv:1907.10516* (2019).

[36] R. L. Plackett. 1975. The analysis of permutations. *Applied Statistics* 24 (1975), 193–202.

[37] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *arXiv preprint arXiv:1306.2597* (2013).

[38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Conference on Uncertainty in Artificial Intelligence (2009)*. 452–461.

[39] Ashkan Rezaei, Rizal Fathony, Omid Memarrast, and Brian Ziebart. 2019. Fair Logistic Regression: An Adversarial Perspective. *arXiv preprint arXiv:1903.03910* (2019).

[40] PAUL ROSENBAUM and Donald Rubin. 1983. The Central Role of the Propensity Score in Observational Studies For Causal Effects. *Biometrika* 70 (1983), 41–55.

[41] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311, 5762 (2006), 854–856.

[42] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2018)*. 2219–2228.

[43] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. *arXiv preprint arXiv:1902.04056* (2019).

[44] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (2016)*. 115–124.

[45] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8 (1992), 229–256.

[46] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *International Conference on Scientific and Statistical Database Management (2017)*. 22.

[47] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *Conference on Information and Knowledge Management (2017)*. ACM, 1569–1578.

[48] Meike Zehlike and Carlos Castillo. 2018. Reducing disparate exposure in ranking: A learning to rank approach. *arXiv preprint arXiv:1805.08716* (2018).